Robust Goal-oriented Behavior in Surprising Environments

An ICAPS 2005 Position Paper

Marshall Brinn, Mark Burstein, Robert Bobrow
BBN Technologies
{mbrinn, mburstein, rusty}@bbn.com

Abstract

Software systems are often vulnerable to failure when confronting unforeseen circumstances. Considering that real world ‘open’ environments are endlessly varied and dynamic, making systems robust in such environments is particularly challenging: the set of conditions cannot be enumerated and handled at design time. Our analysis of different classes of surprise suggests an architecture for achieving robust goal-oriented behavior in open environments. In particular, we content that by adding a reflective layer to standard OODA-loop control processing, recovery from a variety of unanticipated events can be achieved. We present here the elements of that architecture and some results from applying this approach to an example domain.

1. Overview

The nature of software design is that systems are built and operate with particular expectations of the environment in which they will run. Within the constraints of these expectations, the system may perform as desired. However, software systems are notoriously brittle in facing the unanticipated. The behavior may range from the inefficient to the unreasonable to outright failure. This vulnerability stems naturally from the inability on the part of the designer to anticipate every eventuality of an open environment. Despite this, what is desired, at the very least, is a system that behaves reasonably in that it can:

- Not fail the first time it encounters something unanticipated
- Not be surprised in similar circumstances thereafter
- Improve response in each subsequent encounter

Surprise results from encountering observations significantly counter to expectations. A prerequisite, then, for coping with surprise is the ability to be surprised, and thus to have expectations. Such a difference between expectation and observation may be categorized in terms of:

- *Quantitative*: The observation represents a low-probability contingency not explicitly planned for.
- *Qualitative*: The observation represents a zero-probability event, contradicting the current world model.

In either case, some inadequacy in the current model may be the source of the surprise. In the quantitative case, the environment may have changed in some way not yet reflected in models. In the qualitative case, the model may be limited in its scope by certain assumptions, precluding the representation of some events.

In seeking behaviors that are reasonable by the above criteria, we contend that systems must have some flexibility in order to adapt. They must support multiple goals to allow trade-offs of different courses of actions. They must, further, be able to reliably predict the state of the world (as a result of and independent of its actions) in order to plan and act successfully. In the face of surprise, they must be able to monitor and correct their own performance and predictions. Moreover, they must make their model assumptions explicit, to allow for identifying and recovering from the sources of surprising events by questioning these assumptions.

The architectural approach described below seeks to provide software with these attributes in order to enable them to deal reasonably in open environments.

2. Architectural Approaches

The standard cognitive control loop seeks to take actions expected achieve its goals. That is, it relies on some predictive capability to project expected outcomes of potential actions and evaluate the utility of these projections. Underlying this control loop is another process of achieving reliable prediction. A two-tier control mechanism is suggested as illustrated in Figure 1:

- *Environmental Controller (EC)*: Work to achieve goals by acting in world based on predictions
- *Cognitive Controller (CC)*: Work on world model to improve predictions.

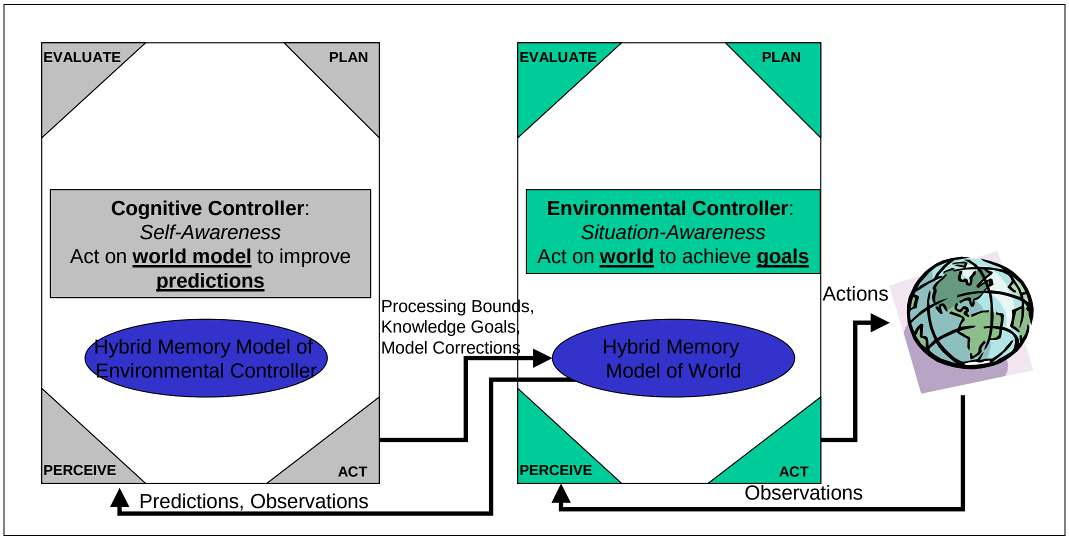

Figure 1: The Cognitive Controller (CC) provides a reflective layer on top of the Environmental Controller (EC), updating its predictive models and providing auxiliary 'knowledge discovery' goals to allow the EC to better operate in the world.

The EC represents a standard OODA loop, developing and acting on plans evaluated from a set of goals and a predictor. We have developed a generic EC that contains a genetic algorithm searching the space of plans (sets of future actions) evaluated to have the best predicted utility.

The CC represents a reflective layer on top of the EC, monitoring its predictions and observations and seeking to improve the EC's predictor accordingly. The CC structure is parallel to that of the EC, but with different goals, sensors and actions. Whereas the EC observes, plans and acts in the world, the CC observes the EC and plans and acts to improve the EC's predictive models.

It also may take control actions on the processing parameters of the EC, manipulating resource and time constraints to allow the EC to operate more effectively in the current environment. The CC is also built on a genetic algorithm; however, it searches the space of qualitative and quantitative changes to the current predictor to better match the recent and long-term observations. The CC is provided with some rough default qualitative models of the world. Initially, it seeks to fill in the quantitative details of these models by fitting to observation data. In addition, however, it seeks to expand these qualitative models by questioning their assumptions systematically to determine possible explanations for anomalous observations, be they significant deviations between observation and expectation or violations of the current qualitative model. We call the parameters by which these qualitative models may be expanded a qualitative metaphysics, containing a broad set of possible contingencies that are initially precluded. Occam's razor encourages us to try to fit the existing data into simple explanations; however, the model should be expanded (temporarily or permanently) by a particular parameter provides a better fit to the observation. For example, by default we assume all actuators and sensors are operating effectively. However, we allow the CC to search models in which the possibility that they may be broken or biased in some way to be considered.

In order for the CC to do its job, it needs adequate data flowing from the CC on actions and their observed outcomes. To that end, the CC will want to not only provide updates to the predictor, but to encourage the EC to take actions that will help the CC to improve its predictor. The relationship of the EC and the CC is one of influence rather than command, and thus it passes 'auxiliary goals' to the EC to include into its overall evaluation of prospective plans. For example, the

CC should not direct the EC to turn on a given actuator; rather it may ask the EC to view more favorably plans that exercise a given actuator, but only if it falls into the broader set of EC goals.

We have developed a generic test framework in which to develop and test robust behavior in open environments. The test framework contains a world simulator that allows for scripted changes to the world and measurements of the world as available to the EC.

3. Example Domain: Temperature Control

To illustrate our approach, we have implemented a system controlling the temperature of a room to desired levels. The components of the domain include:

- *Goals*: Comfort: Maintain the room temperature to as close to 60 degrees as possible, Economy: Minimize time of having AC/Heat running, Simplicity: Minimize the number of times we switch AC/Heat on/off
- *Sensors*: Measurements are available of temperature inside the room and outside the building
- *Controls*: Two heaters are available (one stronger than the other) and one AC

Initially, the sensors are presumed to work, but the qualitative metaphysics understand they may be noisy, biased, or broken. The actuators are presumed to work with qualitative effect that being on raises/lowers room temperature (by how much, how fast are quantitative parameters to be learned and adjusted over time).

At system startup, the CC does not have enough information about the behaviors of the actuators to make a reliable predictor and encourages the EC to sample the space of actuators. The EC obliges since it can’t find any plan that has a good predicted outcome. Once some measurements are available for the actuators, a reasonable predictor is available, and the EC is able to establish classic ‘saw tooth’ control. Soon, the CC realizes that by including the outside temperature (originally ignored for simplicity) in its predictive model, it provides a much better correlation to observed data. The world simulator is subsequently scripted to include different surprising events, including very noisy temperature measurements (which the CC handles by increasing sampling and smoothing) and a broken heater (which the CC learns to predict will have no effect, and the EC will cease to select it in its plans).

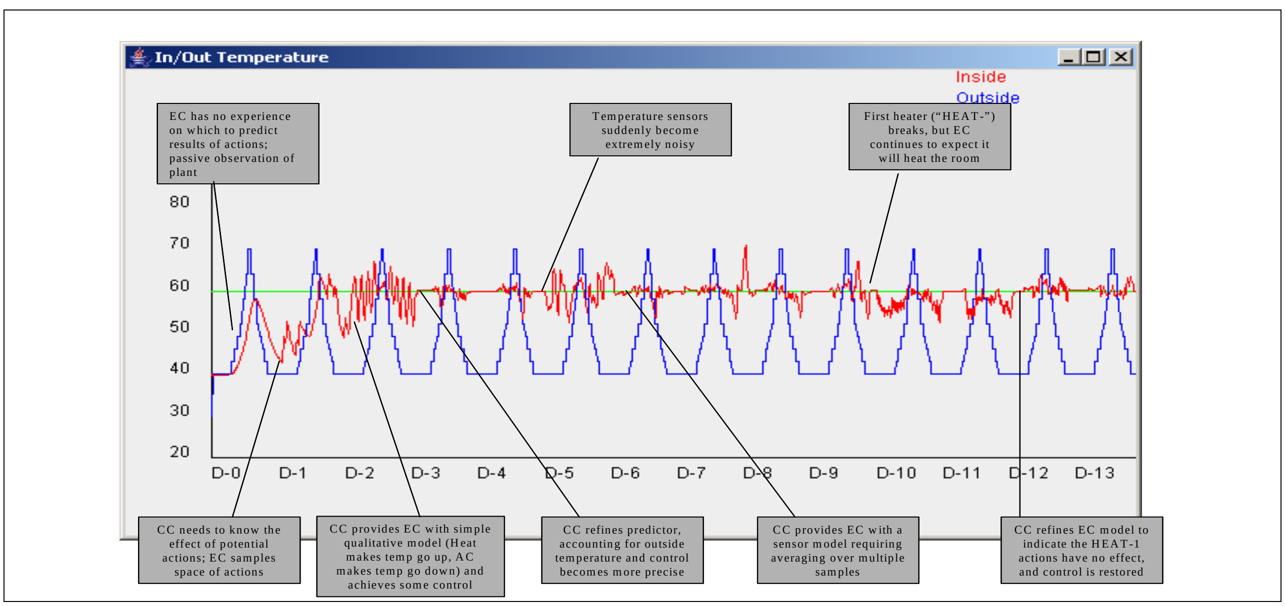

Figure 2: Results from the temperature control simulation, showing the CC providing updated predictors to allow the EC to continue to control temperature in the face of surprising events (e.g. noisy sensors and broken actuators).

Figure 2 illustrates a particular scenario in the life of this temperature control domain, including responses to these and other scenarios.

4. Conclusion

Although we are still in the early stages of exploring this paradigm, we anticipate that hybrid memory and reasoning models will provide additional degrees of robustness to different kinds of surprising, dynamic conditions. In particular, by appropriate layering of case-based, statistical and rule-based/logical models, we contend that the vulnerabilities of one approach may be covered by the strength of another. Further, we are actively investigating incorporating models of uncertainty and lack of knowledge into these predictive models. Specifically, we are seeking to formalize the notion of 'knowledge actions' as another potential action for the EC to take that will provide more information, allowing it and the CC to make better predictions. Additionally, we are investigating how the CC can work to manage the allocation of resources (particularly time) to the EC, in terms of optimal times for planning relative to unfolding execution.

These future efforts not withstanding, several aspects of our approach are suggestive towards a general approach to robust handling of surprise:

- While prediction provides a key to good control, there is a notion of 'good enough prediction' within which reasonable control and response can be achieved (if subject to subsequent fine-tuning).
- Environmental controllers should ascribe benefits to their actions not only by the utility of their predicted outcome, but the information they will provide a reflective layer to allow them to make better predictions in the future.
- The distinction between qualitative and quantitative provides a strong foundation for handling different kinds of surprises.

5. Acknowledgements

This work was performed under DARPA/IPTO contract HR0011-04-C-0078.